Workshop 5

DISTRIBUTED SYSTEMS

An example of request-reply protocols

method	*URL or pathname*	*HTTP version*	*headers*	*message body*
GET	http://www.dcs.qmul.ac.uk/index.html	HTTP/ 1.1		

1. Briefly discuss three abstractions provided by middleware

- Remote Object Invocation
- Remote Procedure Invocation
- Publishing and subscription

2. Explain why a pointer to a local object should not normally be passed to a remote process.

- No assumption of shared memory
  - Unlike a local procedure call, which commonly uses the call-by-reference passing mechanism for input/output parameters, RPCs with input/output parameters have copy-in, copy-out semantics due to the **differing address spaces** of calling and called procedures.

3. Explain how one goes about using Java RMI to build a distributed system. Discuss what needs to be compiled and important aspects of the system that must be used. Actual code is not required.

- *Defining the interface for remote objects* - Interface is defined using the interface definition mechanism supported by the particular RMI software.
- *Compiling the interface* - Compiling the interface generates the proxy, dispatcher and skeleton classes.
- *Writing the server program* - The remote object classes are implemented and compiled with the classes for the dispatchers and skeletons. The server is also responsible for creating and initializing the objects and registering them with the binder.
- *Writing client programs* - Client programs implement invoking code and contain proxies for all remote classes. Uses a binder to lookup for remote objects.

4. Explain the following invocation semantics: maybe invocation, at-least-once, at-most-once

Fault tolerance measures			*Call semantics*
Retransmit request message	*Duplicate filtering*	*Re-execute procedure or retransmit reply*	
No	Not applicable	Not applicable	*Maybe*
Yes	No	Re-execute procedure	*At-least-once*
Yes	Yes	Retransmit reply	*At-most-once*

Book pages 198-199

5. Give two reasons why complete transparency is not always desirable in middleware.

- remote invocations being more prone to failure due to network and remote machines
- latency of remotes invocations is significantly higher than that of local invocations

Read Book 199-200 : Transparency – The discussion is very important

6. Give a reason why Remote Procedure Call is different to Remote Method Invocation, other than procedures and being called rather than methods.

- *Remote method invocation* (RMI) is closely related to RPC but extended into the world of distributed objects. In RMI, a calling object can invoke a method in a potentially remote object. As with RPC, the underlying details are generally hidden from the user.

7. Explain two defining characteristics of an event-based distributed system.

- ***Heterogeneity*:** When event notifications are used as a means of communication, components in a distributed system that were not designed to interoperate can be made to work together. All that is required is that event-generating objects publish the types of events they offer, and that other objects subscribe to patterns of events and provide an interface for receiving and dealing with the resultant notifications.
- ***Asynchronicity*:** Notifications are sent asynchronously by event-generating publishers to all the subscribers that have expressed an interest in them to prevent publishers needing to synchronize with subscribers – publishers and subscribers need to be decoupled.

Read book 244-245